RECEIVED
CENTRAL FAX CENTER
OCT 15 2004

Yee & Associates, P.C.

13760 Noel Road
Suite 900
Dallas, Texas 75240

Main No. (972) 367-2001
Facsimile (972) 367-2008

Facsimile Cover Sheet

To: Commissioner for Patents for **Examiner Hau H. Nguyen** **Group Art Unit 2676**	Facsimile No.: **703/872-9306**
From: Carrie Parker Legal Assistant to Wayne P. Bailey	No. of Pages Including Cover Sheet: **26**
Message: Enclosed herewith: • Transmittal Document; and • Appeal Brief.	
Re: Application No. **09/773,188** Attorney Docket No: **AUS920000757US1**	
Date: Friday, October 15, 2004	
Please contact us at (972) 367-2001 if you do not receive all pages indicated above or experience any difficulty in receiving this facsimile.	*This Facsimile is intended only for the use of the addressee and, if the addressee is a client or their agent, contains privileged and confidential information. If you are not the intended recipient of this facsimile, you have received this facsimile inadvertently and in error. Any review, dissemination, distribution, or copying is strictly prohibited. If you received this facsimile in error, please notify us by telephone and return the facsimile to us immediately.*

PLEASE CONFIRM RECEIPT OF THIS TRANSMISSION BY FAXING A CONFIRMATION TO 972-367-2008.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Nguyen et al.**

Serial No.: **09/773,188**

Filed: **January 31, 2001**

For: **Method and Apparatus for Managing Texture Memory in a Data Processing System**

Group Art Unit: **2676**

Examiner: **Nguyen, Hau H.**

Attorney Docket No.: **AUS920000757US1**

35525
PATENT TRADEMARK OFFICE
CUSTOMER NUMBER

Certificate of Transmission Under 37 C.F.R. § 1.8(a)
I hereby certify this correspondence is being transmitted via facsimile to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, facsimile number (703) 872-9306 on October 15, 2004.

By: Carrie Parker
Carrie Parker

TRANSMITTAL DOCUMENT

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:
ENCLOSED HEREWITH:

- Appeal Brief (37 C.F.R. 41.37).

A fee of $340.00 is required for filing an Appellant's Brief. Please charge this fee to IBM Corporation Deposit Account No. 09-0447. No additional fees are believed to be necessary. If, however, any additional fees are required, I authorize the Commissioner to charge these fees which may be required to IBM Corporation Deposit Account No. 09-0447. No extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and I authorize the Commissioner to charge any fees for this extension to IBM Corporation Deposit Account No. 09-0447.

Respectfully submitted,

Cathrine K. Kinslow
Registration No. 51,886
Duke W. Yee
Registration No. 34,285
YEE & ASSOCIATES, P.C.
P.O. Box 802333
Dallas, Texas 75380
(972) 367-2001
ATTORNEYS FOR APPLICANTS

RECEIVED
CENTRAL FAX CENTER
OCT 15 2004

DOCKET NO. AUS920000757US1 *PATENT*

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Nguyen et al.**	§	
	§	Group Art Unit: **2676**
Serial No. **09/773,188**	§	
	§	Examiner: **Nguyen, Hau H.**
Filed: **January 31, 2001**	§	
	§	
For: **Method and Apparatus for Managing Texture Memory in a Data Processing System**	§	

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Certificate of Transmission Under 37 C.F.R. § 1.8(a)
I hereby certify this correspondence is being transmitted via facsimile to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, facsimile number (703) 872-9306 on October 15, 2004.

By: Carrie Parker
Carrie Parker

APPEAL BRIEF (37 C.F.R. 41.37)

This brief is in furtherance of the Notice of Appeal, filed in this case on August 18, 2004.

The fees required under § 41.20(B)(2), and any required petition for extension of time for filing this brief and fees therefore, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

REAL PARTY IN INTEREST

The real party in interest in this appeal is the following party: International Business Machines Corporation.

RELATED APPEALS AND INTERFERENCES

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

Claims in the application are: 1-30

B. STATUS OF ALL THE CLAIMS IN APPLICATION

1. Claims canceled: 19
2. Claims withdrawn from consideration but not canceled: none
3. Claims pending: 1-18 and 20-30
4. Claims allowed: none
5. Claims rejected: 1-18 and 20-30

C. CLAIMS ON APPEAL

The claims on appeal are: Claims 1-18 and 20-30.

STATUS OF AMENDMENTS

No amendment after final was filed with respect to the present application.

SUMMARY OF CLAIMED SUBJECT MATTER

A. CLAIM 1 - INDEPENDENT

Claim 1 is directed to a memory management technique for graphical texture objects. A stored texture object is selectively removed from memory in response to an inability to allocate sufficient memory to a current texture object. The allocating and selective removal steps are repeated until the current texture object is successfully allocated. If all stored objects are removed from memory without successful allocation to the current texture object, stored texture objects handled by another texture manager are similarly selectively removed. Thus, the claimed invention is directed to selective removal of texture objects in two types of memory resources (a first and second memory resource), where responsive to there being no more stored texture objects in the first memory resource, the selective removal of objects in this first memory resource is halted and stored texture objects are selectively removed from the second memory resource. The present invention thus provides an improved texture memory partitioning and management scheme, thus reducing total memory resource requirements that previously existing in graphical systems.

A flowchart of the above described process is shown in Figure 4 of the present application at reference numbers 400-422, and described at page 11, line 12 - page 14, line 14.

B. CLAIM 11 - INDEPENDENT

Claim 11 is directed to a memory management system for a first memory resource (Figure 3, element 306) and a second memory resource (Figure 3, element 308), where the first memory resource has at least one stored texture object. The memory management system includes a memory allocation unit (Figure 3, element 304) that allocates memory to texture objects in the first memory resource and the second memory resource. The memory management system also includes a texture management system (Figure 3, elements 300 and 302) that tracks allocation of memory for all texture objects, removes a stored texture object within the first memory resource in response to detecting an inability to allocate memory to a current texture object in the first memory resource, calls the memory allocation unit to allocate memory to the current texture

object after the stored texture object is removed, and continues to remove texture objects from the first memory resource until sufficient memory is allocated to the current texture object, as described at Applicants' specification at page 9, line 30 – page 11, line 11.

C. CLAIM 15 - INDEPENDENT

See above summary regarding Claim 1.

D. CLAIM 20 - INDEPENDENT

See above summary regarding Claim 1.

E. CLAIM 30 - INDEPENDENT

See above summary regarding Claim 1.

F. CLAIM 20 – 35 USC 112, 6th PARAGRAPH MEANS-PLUS-FUNCTION

Corresponding structure shown at element 100 in Figure 1 and element 200 in Figure 2, as described at Specification page 6 line 4 – page 9 line 29, for all recited elements in Claim 20.

G. CLAIM 25 – 35 USC 112, 6th PARAGRAPH MEANS-PLUS-FUNCTION

Corresponding structure shown at element 100 in Figure 1 and element 200 in Figure 2, as described at Specification page 6 line 4 – page 9 line 29, for all recited elements in Claim 25.

H. CLAIM 27 – 35 USC 112, 6th PARAGRAPH MEANS-PLUS-FUNCTION

Corresponding structure shown at element 100 in Figure 1 and element 200 in Figure 2, as described at Specification page 6 line 4 – page 9 line 29, for all recited elements in Claim 27.

I. CLAIM 28 – 35 USC 112, 6th PARAGRAPH MEANS-PLUS-FUNCTION

Corresponding structure shown at element 100 in Figure 1 and element 200 in Figure 2, as described at Specification page 6 line 4 – page 9 line 29, for all recited elements in Claim 28.

J. CLAIM 29 – 35 USC 112, 6th PARAGRAPH MEANS-PLUS-FUNCTION

Corresponding structure shown at element 100 in Figure 1 and element 200 in Figure 2, as described at Specification page 6 line 4 – page 9 line 29, for all recited elements in Claim 29.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. GROUND OF REJECTION 1 (Claims 1-11, 13-18 and 20-23)

Claims 1-11, 13-18 and 20-23 stand rejected under 35 U.S.C. § 103 as being unpatentable over Gannett (U.S. Patent No. 5,790,130) in view of Sethi et al. (U.S. Patent No. 6,600,493).

B. GROUND OF REJECTION 2 (Claim 12)

Claim 12 stands rejected under 35 USC § 103 as being unpatentable over Gannett (U.S. Patent No. 5,790,130) in view of Sethi et al. (U.S. Patent No. 6,600,493) further in view of Saunders (U.S. Patent No. 5,917,497).

ARGUMENT

A. GROUND OF REJECTION 1 (Claims 1-11, 13-18 and 20-23)

A.1. Group I Claims 1-5, 7, 15-18, 20-24, 26, 30

Generally speaking, the present invention is directed to management/control of two memory resources – a first memory resource and a second memory resource – for use in maintaining texture objects. Memory is allocated to a current texture object in the first memory resource. In response to an inability to allocate sufficient memory to the current texture object, a stored texture object *in the first memory resource is selectively removed*. The allocating and selectively removing steps are repeated, if there is another stored texture object present in the first memory resource, until the current texture object is allocated sufficient memory. This repeating step is halted in response to an absence of any stored texture objects being present in the first memory resource. Responsive to the halting step, stored texture objects *in a second memory resource are selectively removed* if an inability to allocate sufficient memory to the current texture object is present. In response to selectively removing stored texture objects, memory in the second memory resource is allocated to the current texture object. In rejecting Claim 1, the Examiner has only alleged removal in one type of memory (per Gannet), and allocation in two types of memory (per Sethi), as further detailed below. The Examiner has therefore failed to establish a prima facie showing with respect to Group I[1], and thus the burden has not shifted to Appellants to rebut such obviousness assertion[2]. In addition, as a prima facie case of obviousness has not been established by the Examiner, the rejection of Group I is shown to be in error[3].

In rejecting Claim 1, the Examiner equates Gannett's teaching of shared memory location as

---

[1] In rejecting claims under 35 U.S.C. Section 103, the examiner bears the initial burden of presenting a prima facie case of obviousness. *In re Oetiker*, 977 F.2d 1443, 1445, 24 USPQ2d 1443, 1444 (Fed. Cir. 1992). To establish prima facie obviousness of a claimed invention, all of the claim limitations must be taught or suggested by the prior art. MPEP 2143.03. *See also, In re Royka*, 490 F.2d 580 (C.C.P.A. 1974).

[2] Only if that burden is met, does the burden of coming forward with evidence or argument shift to the applicant. *In re Oetiker*, supra.

[3] If the examiner fails to establish a prima facie case, the rejection is improper and will be overturned. *In re Fine*, 837 F.2d 1071, 1074, 5 USPQ2d 1596, 1598 (Fed. Cir. 1988).

the claimed first memory resource, and Gannett's teaching of TIM's own allocated system software memory location as reading on the claimed second memory resource. The shared memory is shown by element 852 of Gannet's Figure 30 (as described by Gannet at Col. 69, lines 3-10), and the TIM allocated system software memory is shown by element 854 of Gannet's Figure 30 (as described by Gannet at Col. 68, lines 52-55). Importantly, the only texture items that get removed are the one's contained within the hardware driver data storage 851 of Figure 30 (which is neither the alleged first memory resource nor the alleged second memory resource), as described by Gannett at Col. 8, lines 17-28; Col. Col. 8, line 66 – Col. 9, line 20; Col. 10, line 48 – Col. 11, line 4; Col. 14, lines 10-65; Col. 42, lines 6-21; Col. 64, lines 22-35. Gannet does not teach or otherwise suggest selective removal of a stored texture object in both (1) the shared memory (Gannet Figure 30, element 852 which is alleged by the Examiner to read on the claimed first memory resource), AND (2) the TIM memory (Gannet Figure 30, element 854 which is alleged by the Examiner to read on the claimed second memory resource). In summary, Gannet teaches removal of objects in a single memory, whereas the claimed invention of Group I is directed to removal of texture data objects in two memory resources – a first memory resource and a second memory resource.

Nor do the teachings of the cited Sethi reference overcome this teaching deficiency. Sethi teaches a memory allocation scheme, where if an available amount of memory exists in graphics memory to satisfy a request, the memory allocation process ends. If there is not sufficient available graphics memory, available portions of system memory 21 are allocated to make up for the deficit amount of graphics memory (Sethi Col. 2, lines 55-60). Thus, Sethi teaches memory allocation using two memory resources, but does not teach or otherwise suggest selective removal of a stored texture object from two memory resources, as claimed. Thus, the teachings of Sethi do not overcome the above described deficiency in the Gannet teachings.

Further with respect to Group I, none of the cited references teach or suggest that removal of a texture object from the first memory resource is responsive to an inability to allocate sufficient memory to the current texture object, whereas the removal of a texture object from the second memory resource is responsive to halting of a repeated allocating and selective removal with respect to the first memory resource. Nor has the Examiner alleged such two-step selective removal with differing responsive conditions. Thus, Appellants further show a failure by the Examiner to establish a prima facie showing of obviousness with respect to Group I.

In summary, none of the cited references teach or suggest *selective removal* of texture

objects from *two memory resources* – a first memory resource and a second memory resource – the selective removal in the first memory resource being *responsive to an inability to allocate sufficient memory to the current texture object*, and the selective removal in the second memory resource being *responsive to* halting of a repeated allocating and selective removal with respect to the first memory resource. Claim 1 expressly recites "selectively removing a stored texture object in the first memory resource in response to an inability to allocate sufficient memory to the current texture object" AND "*responsive to the halting step, selectively removing stored texture objects in a second memory resource if an inability to allocate sufficient memory to the current texture object is present*". None of the cited references teach or suggest this second selective removal step, nor has the Examiner alleged any such teaching of two-step selective removal. The Examiner merely alleges that *data will be stored* in either a first memory resource or a second memory resource depending on certain conditions being met. In contrast, Claim 1 is direct to *selective removal* of data from two memory resources depending on certain other conditions being met. Thus, it is shown in detail (above) that a prima facie case of obviousness has not been made with respect to Group I, and accordingly the burden has not shifted to Appellants to rebut an obviousness assertion with respect to the Group I claims.

A.2. Group II Claims 6 and 25

Appellants initially show error in the Group II claim rejection for reasons given above with respect to the Group I claims. Further with respect to Group II (Claims 6 and 25), Appellants urge that none of the cited references teach or suggest that the first memory resource is assigned to a client application. In rejecting Claims 1-6, the Examiner equates the claimed first memory resource with Gannet's shared memory. Appellants urge that this shared memory is not assigned to a client application, as claimed. Rather, this shared memory is shared between the graphics API, graphics hardware driver and the TIM daemon (Col. 9, lines 30-34). None of the cited references teach or suggest, nor has the Examiner alleged any teaching or suggestion, that the claimed first memory resource is assigned to a client application. Therefore, a prima facie case of obviousness has not been made with respect to the Group II claims, and accordingly the burden has not shifted to Appellants to rebut an obviousness assertion with respect to the Group II claims.

A.3. Group III Claims 8 and 27; A.4. Group IV Claims 9 and 28

Appellants initially show error in the Group III claim rejection for reasons given above with respect to the Group I claims. Further with respect to Group III (Claims 8 and 27), none of the cited references teach or suggest selectively removing *all* stored texture objects in the second memory resource in response to an inability to allocate sufficient memory to the current texture object; and further with respect to Group IV (Claims 9 and 28), none of the cited references teach or suggest selectively removing *a single* stored texture objects in the second memory resource in response to an inability to allocate sufficient memory to the current texture object. As can be seen, the claims of Group III and IV recite particular implementations of the selectively removing (stored) texture objects step recited in Claim 1. The Examiner appears to gloss over the distinctions between Claims 8 and 9, as Claim 8 recites that all stored texture objects are removed in the second memory resource, whereas Claim 9 recites that one stored texture object is removed in the second memory resource. The Examiner has failed to establish that both of these specific implementations are taught or suggested by the cited references. The Examiner cites Sethi Col. 4, lines 19-31 as teaching "removing all objects that have been used". Appellants show that such assertion does not establish that this removal is *responsive to a halting step*, as expressly recited in the combination of Claim 1 and Claim 8, or that this selective removal is in *response to an inability to allocate* sufficient memory. Quite to the contrary, this clean-up of memory by Sethi is responsive to a *successful allocation* of memory and subsequent object processing.

In addition, such cited passage does not establish that all (Group III) or one (Group IV) stored texture object(s) is/are removed in the second memory, only that unused graphics memory is released to a pool. Therefore, a prima facie case of obviousness has not be established with respect to Groups III and IV for at least the many reasons listed above, and the claims of such groups are thus shown to have been erroneously rejected.

A.5. Group V Claims 10 and 29

Appellants initially show error in the Group V claim rejection for reasons given above with respect to the Group I claims. Further with respect to the claims of Group V (Claims 10 and 29), Appellants show that none of the cited references teach or suggest an identifier that *identifies a memory resource* in which the (removed) texture object was located. Nor has the Examiner alleged

any such teaching or suggestion. Therefore, a prima facie case of obviousness has not been established with respect to Group V, and the claims of such group are thus shown to have been erroneously rejected.

A6. Group VI Claims 11 and 14

With respect to Group VI (Claims 11 and 14), Claim 11 recites both a memory allocation unit and a texture management system, where the texture management system tracks allocation of memory for all texture objects, removes a stored texture object within the first memory resource in response to detecting an inability to allocate memory to a current texture object, calls the memory allocation unit to allocate memory to the current texture object after the stored texture object is removed, and continues to remove texture objects from the first memory resource until sufficient memory is allocated to the current texture object. The Examiner has not alleged any teaching or suggestion by any of the cited references of the claimed "memory allocation unit" and "texture management system", where the texture management system calls the memory allocation unit after a stored texture object is removed. Thus, the Examiner has failed to establish a prima facie showing of obviousness with respect to Group VI, and therefore the burden has not shifted to Appellants to rebut the obviousness rejection.

A.7. Group VII Claim 13

Appellants initially show error in the Group VII claim rejection for reasons given above with respect to the Group VI claims. Further with respect to Group VII (Claim 13), Claim 13 recites a claimed feature of "wherein the texture management system removes texture objects from the second memory resource if stored texture objects are absent from the first memory resource and insufficient memory has been allocated to the current memory object". As can be seen, the removal of texture objects from the second memory resource is conditioned upon two conditions being met – the stored texture objects being absent from the first memory resource AND insufficient memory has been allocated to the current memory object. The Examiner has not alleged any teaching or suggestion of removal of texture objects from the second memory resource based on these two conditions being met, and thus the Examiner has failed to establish a prima facie showing of obviousness with respect to Group VII.

B. GROUND OF REJECTION 2 (Claim 12)

B.1. Group VIII Claim 12

Appellants initially show error in the Group VIII claim rejection for reasons given above with respect to the Group VI claims. Further with respect to Group VIII (Claim 12), Claim 12 recites a claimed feature of "wherein the texture management system returns an error if stored texture objects are absent from the first memory resource and insufficient memory has been allocated to the current texture object". As can be seen, the error that is returned is conditioned upon two conditions being met – the stored texture objects being absent from the first memory resource AND insufficient memory has been allocated to the current texture object. The Examiner has not alleged any teaching or suggestion of returning an error based on these two conditions being met, but rather has merely alleged that the cited Saunders reference teaches returning an error "if it is determined that sufficient memory could not be allocated". Thus, the Examiner has failed to establish a prima facie showing of obviousness with respect to Group VIII.

Conclusion

Thus, Appellants have shown that the rejection of Claims 1-18 and 20-30 is in error, and requests that such rejection by reversed by the Board, and further requests that such claims be allowed.

Cathrine K. Kinslow
Reg. No. 51,886
Wayne P. Bailey
Reg. No. 34,289
YEE & ASSOCIATES, P.C.
PO Box 802333
Dallas, TX 75380
(972) 367-2001

CLAIMS APPENDIX

The text of the claims involved in the appeal are:

1. A method in a data processing system for managing a set of memory resources used to store texture objects, the data processing system comprising a first memory resource with at least one stored texture object, the method comprising:

allocating memory to a current texture object in the first memory resource within the set of memory resources;

selectively removing a stored texture object in the first memory resource in response to an inability to allocate sufficient memory to the current texture object;

repeating the allocating and selectively removing steps if there is another stored texture object present in the first memory resource until the current texture object is allocated sufficient memory, wherein the repeated selectively removing step is with respect to the another stored texture object ;

halting the repeating step in response to an absence of any stored texture objects being present in the first memory resource;

responsive to the halting step, selectively removing stored texture objects in a second memory resource if an inability to allocate sufficient memory to the current texture object is present; and

allocating memory in the second memory resource to the current texture object in response to selectively removing stored texture objects.

2. The method of claim 1, wherein the first memory resource is system memory.

3. The method of claim 1, wherein the first memory resource is an advanced graphic port memory.

4. The method of claim 1, wherein the second memory resource is a frame buffer.

5. The method of claim 1, wherein the secondary memory resource is assigned to a kernel application.

6. The method of claim 1, wherein the first memory resource is assigned to a client application.

7. The method of claim 1, wherein the stored texture object is a texture object used less than a threshold value.

8. The method of claim 1, wherein the step of selectively removing texture objects comprises:

selectively removing all stored texture objects in the second memory resource in response to an inability to allocate sufficient memory to the current texture object.

9. The method of claim 1, wherein the step of selectively removing texture objects comprises:

selectively removing a single stored texture object in the second memory resource in response to an inability to allocate sufficient memory to the current texture object.

10. The method of claim 1, wherein an identifier is associated with a texture object when the texture object is removed from the set of memory resources in which the identifier identifies a memory resource in which the texture object was located within the set of memory resources.

11. A memory management system for a first memory resource and a second memory resource, the first memory resource having at least one stored texture object, comprising:

a memory allocation unit, wherein the memory allocation unit allocates memory to texture objects in the first memory resource and the second memory resource;

a texture management system, wherein the texture management system tracks allocation of memory for all texture objects, removes a stored texture object within the first memory resource in response to detecting an inability to allocate memory to a current texture object in the first memory resource, calls the memory allocation unit to allocate memory to the current texture object after the stored texture object is removed, and continues to remove texture objects from the first memory resource until sufficient memory is allocated to the current texture object.

12. The memory management system of claim 11, wherein the texture management system returns an error if stored texture objects are absent from the first memory resource and insufficient memory has been allocated to the current texture object.

13. The memory management system of claim 11, wherein the texture management system removes texture objects from the second memory resource if texture objects are absent from the first memory resource and insufficient memory has been allocated to the current memory object.

14. The memory management system of claim 11, wherein the texture management system includes a first texture manager and a second texture manager, wherein the first texture manager track allocation of memory for all texture objects in the first memory resource and wherein the second texture manager track allocation of memory for all texture objects in the second memory resource.

15. A data processing system comprising:

a bus system;

a memory connected to the bus system, wherein a set of instructions are located in the memory; and

a processor unit connected to the bus system, wherein the processor unit executes the set of instructions to allocate memory to a current texture object in a first memory resource within the set of memory resources; selectively remove a stored texture object in the first memory resource in response to an inability to allocate sufficient memory to the current texture object; repeat instructions to the allocate memory and selectively remove a stored texture object until the current texture object is allocated sufficient memory; halt the repeating in response to an absence of any stored texture objects being present in the first memory resource, selectively remove stored texture objects in a second memory resource in response to an inability to allocate sufficient memory to the current texture object in response to halting instructions to repeat; and

allocate memory in the second memory resource to the current texture object in response to selectively removing stored texture objects.

16. The data processing system of claim 15, wherein the bus system includes a primary bus and a secondary bus.

17. The data processing system of claim 15, wherein the processor unit includes a single processor.

18. The data processing system of claim 15, wherein the processor unit includes a plurality of processors.

20. A data processing system for managing a set of memory resources used to store texture objects, the data processing system comprising a first memory resource with at least one stored texture object, the data processing system further comprising:

first allocating means for allocating memory to a current texture object in the first memory resource within the set of memory resources;

first selectively removing means for selectively removing a stored texture object in the first memory resource in response to an inability to allocate sufficient memory to the current texture object;

repeating means for repeating initiation of the first allocating means and first selectively removing means if there is another stored texture object present in the first memory resource

until the current texture object is allocated sufficient memory, wherein the repeated initiation of the first selectively removing means is with respect to the another stored texture object;

halting means for halting the repeating means in response to an absence of any stored texture objects being present in the first memory resource;

second selectively removing means, responsive to the halting step, for selectively removing stored texture objects in a second memory resource if an inability to allocate sufficient memory to the current texture object is present; and

second allocating means for allocating memory in the second memory resource to the current texture object in response to selectively removing stored texture objects.

21. The data processing system of claim 20, wherein the first memory resource is system memory.

22. The data processing system of claim 20, wherein the first memory resource is an advanced graphic port memory.

23. The data processing system of claim 20, wherein the second memory resource is a frame buffer.

24. The data processing system of claim 20, wherein the secondary memory resource is assigned to a kernel application.

25. The data processing system of claim 20, wherein the first memory resource is assigned to a client application.

26. The data processing system of claim 20, wherein the stored texture object is a texture object used less than a threshold value.

27. The data processing system of claim 20, wherein the step of selectively removing texture objects comprises:

third selectively removing means for selectively removing all stored texture objects in the second memory resource in response to an inability to allocate sufficient memory to the current texture object.

28. The data processing system of claim 20, wherein the step of selectively removing texture objects comprises:

fourth selectively means for selectively removing a single stored texture object in the second memory resource in response to an inability to allocate sufficient memory to the current texture object.

29. The data processing system of claim 20, wherein an identifier is associated with a texture object when the texture object is removed from the set of memory resources in which the identifier identifies a memory resource in which the texture object was located within the set of memory resources.

30. A computer program product in a computer readable medium for managing a set of memory resources used to store texture objects, the computer program product comprising:

first instructions for allocating memory to a current texture object in a first memory resource within the set of memory resources;

second instructions for selectively removing a stored texture object in the first memory resource in response to an inability to allocate sufficient memory to the current texture object;

third instructions for repeating the allocating and selectively removing steps if there is another stored texture object present in the first memory resource until the current texture object is allocated sufficient memory, wherein the repeated second instructions for selectively removing is with respect to the another stored texture object;

fourth instructions for halting the repeating step in response to an absence of any stored texture objects being present in the first memory resource;

fifth instructions, responsive to the halting step, for selectively removing stored texture objects in a second memory resource if an inability to allocate sufficient memory to the current texture object is present; and

sixth instructions for allocating memory in the second memory resource to the current texture object in response to selectively removing stored texture objects.

EVIDENCE APPENDIX

There is no evidence to be presented.

RELATED PROCEEDINGS APPENDIX

There are no related proceedings.